

Performance Evaluation of Chord in Mobile Ad Hoc Networks

Curt Cramer
System Architecture Group
Universität Karlsruhe (TH), Germany
cramer@ira.uka.de

Thomas Fuhrmann
System Architecture Group
Universität Karlsruhe (TH), Germany
fuhrmann@ira.uka.de

ABSTRACT

Mobile peer-to-peer applications recently have received growing interest. However, it is often assumed that structured peer-to-peer overlays cannot efficiently operate in mobile ad hoc networks (MANETs). The prevailing opinion is that this is due to the protocols' high overhead cost. In this paper, we show that this opinion is misguided.

We present a thorough simulation study evaluating Chord in the well-known MANET simulator GloMoSim. We found the main issue of deploying Chord in a MANET not to be its overhead, but rather the protocol's pessimistic timeout and failover strategy. This strategy enables fast lookup resolution in spite of highly dynamic node membership, which is a significant problem in the Internet context. However, with the inherently higher packet loss rate in a MANET, this failover strategy results in lookups being inconsistently forwarded even if node membership does not change.

Categories and Subject Descriptors

C2.2 [Computer Systems Organization]: Network Protocols; C.4 [Computer Systems Organization]: Performance of Systems

General Terms

Performance, Experimentation

Keywords

Structured peer-to-peer network, MANET, Chord

1. INTRODUCTION

An ad hoc network is a multi-hop wireless network operating without an infrastructure. That is, all nodes have to cooperatively perform tasks such as routing or service discovery using distributed approaches. The design constraints of ad hoc networks are: 1) infrastructure-less, decentralized operation, 2) autonomous and fast (re-) configuration, 3)

scarce bandwidth, 4) unreliable wireless channels, and 5) node mobility. While there has been published a considerable amount of research on routing under these constraints, the relevance of ad hoc networks can only be justified by the availability of applications.

Distributed applications require basic services such as naming. As a consequence of the absent infrastructure, these services have to be provided by the network itself. Though originally proposed for the large-scale Internet environment, their self-organization capabilities make peer-to-peer (P2P) overlay protocols an attractive choice for implementing the services. Assuming that the items requested from a service are rare, the versatile key-based routing primitive offered by structured P2P overlays such as Pastry [18] and Chord [19] are the more suitable and efficient alternative for implementing the service rather than flooding-based, unstructured overlays.

On account of that, a growing interest in applying structured P2P protocols to the area of mobile ad hoc networks (MANETs) can recently be observed. The communication overhead for maintaining a virtual overlay network in a mobile environment and the performance penalty arising from routing in this virtual overlay are deemed to be the main issues. While some authors rebut the efficiency of structured overlays in ad hoc networks altogether (cf. refs. [2, 9, 11]), and others describe working examples (cf. refs. [17, 21]), no definitive answer has been given yet. Nevertheless, a thorough experimental evaluation of these issues is absent from the literature.

We have evaluated the behavior of the well-known Chord lookup protocol in a MANET using the network simulator GloMoSim [1]. Our results contradict the prevalent opinion that the performance penalty resulting from mismatching virtual and physical network topologies is prohibitive. Rather, as a result of packet loss inherent to wireless environments, Chord's pessimistic failover strategy prematurely discards valuable state. In combination with key replication, this strategy helps mitigating the effects of a highly transient node population as present in the Internet context; however, in a MANET, the strategy results in lookups being inconsistently forwarded even if node membership does not change. Replication can be employed to counter this effect, but is expensive because of the restricted bandwidth.

The remainder of this paper is structured as follows: In the next section, we provide a short survey on literature related to P2P in MANETs. Our performance evaluation of Chord in MANETs is presented and discussed in Section 3. We conclude our findings in Section 4.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiShare'06, September 25, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-558-4/06/0009 ...\$5.00.

2. RELATED WORK

The literature on structured P2P overlays in MANETs predominantly argues that a significant and inhibiting bandwidth shortage arises from the obliviousness of the virtual overlay to the underlying physical network [9,11,12,20]. This argument is used to motivate specialized cross-layer solutions [17,21] or even to advocate the use of unstructured P2P protocols (see e.g. references [2,11]). Despite this strong focus on topology adaptation, little work has been done to quantify its effect, or to identify other issues with the deployment of standard P2P protocols in MANETs and to determine the relative significance of topology adaptation.

Ekta [17] is a cross-layer solution for providing a DHT in MANETs. It is a combination of the proximity-aware Pastry overlay and of DSR. The details of the proximity adaptation in the MANET scenario have not explicitly been evaluated by the authors. MADPastry [21] uses location-dependent addressing to adapt the overlay to the underlying network. The authors compare their specialized solution with a combination of Pastry and AODV. CrossROAD [3] extends OLSR with structured overlay functionality by hashing the IP addresses in the routing table. This is equivalent to creating a full mesh overlay topology, as a node maintains routes to all other nodes in the network. The authors compared CrossROAD's performance with that of Pastry overlaid on OLSR.

VRR [5] and derivatives of the successor maintenance principle laid out by ISPRP [7] (e.g., SSR [10]) are advanced structured P2P protocols for MANETs. While being cross-layer solutions using location-independent addressing, they do not discover routes by flooding.

3. PERFORMANCE EVALUATION

The goal of this work is to identify the key performance issues of deploying a standard P2P protocol on top of a MANET routing protocol. For this purpose we have implemented and evaluated Chord in the MANET simulator GloMoSim. We are particularly interested in Chord's ability to correctly resolve lookups under a variety of conditions (i.e., different network sizes, degrees of node mobility, and amounts of application traffic). The remainder of this section describes the simulation study and its results in detail.

3.1 Simulation Model

All nodes use the IEEE 802.11 DFWMAC with RTS/CTS extension, and a 2.4-GHz radio interface with a transmission speed of 2 Mbps. Radio propagation follows the two-ray model with a nominal transmission range of 250 m.

Our basic scenario consists of a network of 50 nodes uniformly distributed on a square. The area of the square is chosen to yield an average node density of 14 nodes per transmission circle. All nodes move according to the random direction model with node speeds uniformly distributed between 0 m/s and 2 m/s, modeling the motion of pedestrians. The duration of a movement into one direction is drawn from an exponential distribution with a mean of 300 s. Each node issues requests every 5 s on average.

We analyze the respective influence of the parameters of offered load, node mobility, and network size on protocol performance. When varying the value of a parameter, the values of the other parameters remain fixed at their basic values.

Each data point represents an average over ten independent simulation runs and is shown together with an approximate 95%-confidence interval approximated using Student's t distribution. A run simulates two hours of time. To avoid initial transient effects caused by an inconsistent overlay during its bootstrapping phase, the application traffic starts after one hour of simulated time and lasts for the remaining hour. All nodes join the Chord network within the first 50 seconds.

3.1.1 Routing Protocols

As the main subject of our evaluation is Chord, we wanted to factor out any artifacts of using a specific routing protocol. Therefore, we simulated Chord with three different routing protocols: Ad hoc On-Demand Distance Vector (AODV) routing [16], Dynamic Source Routing (DSR) [14], and Optimized Link State Routing (OLSR) [13]. All protocols were configured with the default values of their parameters.

OLSR is a proactive link state routing protocol. It uses the idea of multipoint relaying to reduce the overhead of plain link state routing. Only a subset of all links is advertised, and the efficiency of flooding is increased by only a subset of all nodes retransmitting messages. (GloMoSim implementation by University of Niigata, Japan, follows specification in ref. [6].)

DSR is a reactive source routing protocol. The protocol makes extensive use of caching for reducing the overhead of flooding route discovery messages. Caching also helps finding alternative routes in the event of route breaks. (GloMoSim implementation follows specification in ref. [4].)

AODV is a reactive distance vector routing protocol. Loop-free routing is ensured by employing sequence numbers. (GloMoSim implementation follows specification in ref. [15].)

3.1.2 The Chord Lookup Protocol

We implemented Chord along the lines of the latest published specification in ref. [19]. Chord uses a remote procedure call (RPC) model. The protocol runs on the application layer and sends control messages using the UDP transport protocol. Reliability of message transfers is enabled by the application layer using timers and retransmissions. RPCs are timed out after 500 ms unless an acknowledgment has been received. After three unsuccessful transmission attempts, an RPC is considered a failure.

Key lookups are implemented as recursive RPCs to the *find_successor()* function. Each recursive call is acknowledged to the preceding hop. The lookup reply is directly sent from the final hop to the originator of the lookup. Besides making reliable RPCs to the individual hops, lookups are also timed out on a global scale. If the originator does not receive a reply to its lookup within 15 s, it retransmits the lookup (at most three times).

If a single transmission attempt to the next hop in a lookup times out, the forwarding node assumes that the unreachable node has left the network or has failed. The forwarding node therefore removes all state associated with the unreachable node from its forwarding database. This database is comprised of the finger table and the successor

list. After choosing a new next hop from its database (i.e., the closest remaining node preceding the lookup key), the forwarding node retries the RPC to that node. A lookup terminates when it has reached the successor of the requested key.

Each node reconciles a list of four consecutive successors with its immediate successor every 30 s. We do not replicate objects as this is an application-specific task, and we only consider the performance of the Chord lookup protocol itself.

3.1.3 Flooding-Based Key Lookup

For comparison, we implemented a naïve lookup protocol based on flooding. Key requests are flooded throughout the network in a best-effort manner. The node managing the requested key responds by sending back an answer using the reverse path as recorded in the request message while flooding. This protocol is similar to the resource discovery model described in ref. [17], except for that it does not employ replication or any optimizations such as caching.

As discussed by Zahn and others [21], flooding marks the baseline of key lookup performance: if flooding a lookup costs less than routing it through a Chord network deployed on a flooding-based routing protocol, it is unreasonable to use Chord.

3.1.4 Traffic Model

Traffic is generated by an application protocol called *ReqGen*. The protocol module reads a request profile, that is a file listing tuples that describe the events of nodes requesting keys, and the module injects these requests into the network at the specified times.

When used in conjunction with Chord, a node first resolves the key into the associated node’s network address. The node then sends the request to the ReqGen instance on that node. A recipient of a request only replies if it is responsible for managing the key. This only applies if the prior Chord lookup was consistently resolved.

The flooding lookup protocol does not resolve the key prior to sending the request, but simply floods the request. By sharing the same request profiles between Chord and the flooding-based lookup protocol, ReqGen enables fair comparisons.

The time between two consecutive requests issued by a node is exponentially distributed with a given mean, and requests are sent to uniformly distributed random destinations. We vary the load offered to the network by varying the average interarrival time of the request processes. Request and response packets have a length of 512 bytes each.

3.2 Performance Results

3.2.1 Performance Metrics

We use the following metrics to analyze protocol performance:

Request success ratio The total fraction of requests for which the originator successfully receives an answer. This metric reflects the routing protocol’s ability to deliver packets, Chord’s ability to resolve multi-hop lookups, and most importantly its ability to *consistently* resolve these lookups. In our communication model, a node receiving a ReqGen request only responds when it is responsible for managing the requested key. That is, the metric regards a request

as failed if the preceding key lookup in Chord yields an incorrect result.

Overlay consistency The percentage of all nodes that from a global point of view have selected their correct successor. Consistency is a function of time.

End-to-end delay The cumulated delay between issuing a request and receiving the reply. This includes both the delay of performing the Chord lookup and of the request/response cycle.

Total network load The *total* load exerted on the network. In contrast to the offered load, it comprises application and control traffic, and each transmission on a multi-hop route is separately counted.

3.2.2 Varying Offered Load

In the first part of the experiments, we used a 50-node network with a maximum node speed of 2 m/s and a varying traffic load. For increasing the traffic load, the interarrival time of requests was reduced. Requests were sent at an average per-node rate of 1/15 Hz (offered load: 6 MB), 1/10 Hz (9 MB), 1/5 Hz (18 MB) and 1 Hz (92 MB), respectively.

The request success ratios for the deployments of Chord over the different routing protocols AODV, OLSR, and DSR are quite similar (see Fig. 1(a)). They are approximately 50% for the lowest traffic load. Under increasing load, the ratios degrade to less than 10%. In all cases, the flooding protocol significantly outperforms Chord with over 90% success for the lower traffic loads, and still over 20% for the highest offered load.

Chord’s low success ratios can be explained from the trend in its consistency degree (see Figs. 1(d)–1(f)). The ability to correctly resolve key lookups depends on the consistency. It can be seen from the figures that the time to bootstrap a Chord network is linear in the size of the network (cf. ref. [8]). Out of the three protocol combinations, only Chord/AODV is able to initially achieve close to 100% consistency. In any case, consistency sharply drops and stabilizes at a lower level when application traffic gets injected into the network (after 3600 s).

The average end-to-end delay of a Chord lookup ranges from ca. 750 ms at 1/15 Hz to ca. 220 ms at 1 Hz. The drop in the average end-to-end delay for higher traffic loads can be explained by the decreasing variability. At higher loads, requests that take longer to complete (i.e., requests that traverse longer paths during the lookup), are completed less often than at lower loads. For the flooding protocol, the end-to-end delay increases from 99 ms at a request rate of 1/15 Hz to 113 ms at 1/10 Hz, 194 ms at 1/5 Hz, and 6.2 s at 1 Hz (Fig. 1(b)).

Fig. 1(c) depicts the total load on the network. With flooding, the load steeply increases from 350 MB to 2051 MB. The Chord experiments show a slower increase, for example from 135 MB to 463 MB with Chord over DSR.

3.2.3 Varying Node Mobility

In the second experiment series, we used a 50-node network where each node sent requests every 5 s on average. The maximum node speed was set to 0 m/s for a static network, 2 m/s for a network of moderate mobility, and 5 m/s for a network of high mobility.

It can be seen that the flooding protocol is robust against network mobility. Over 92% of the requests succeed (Fig.

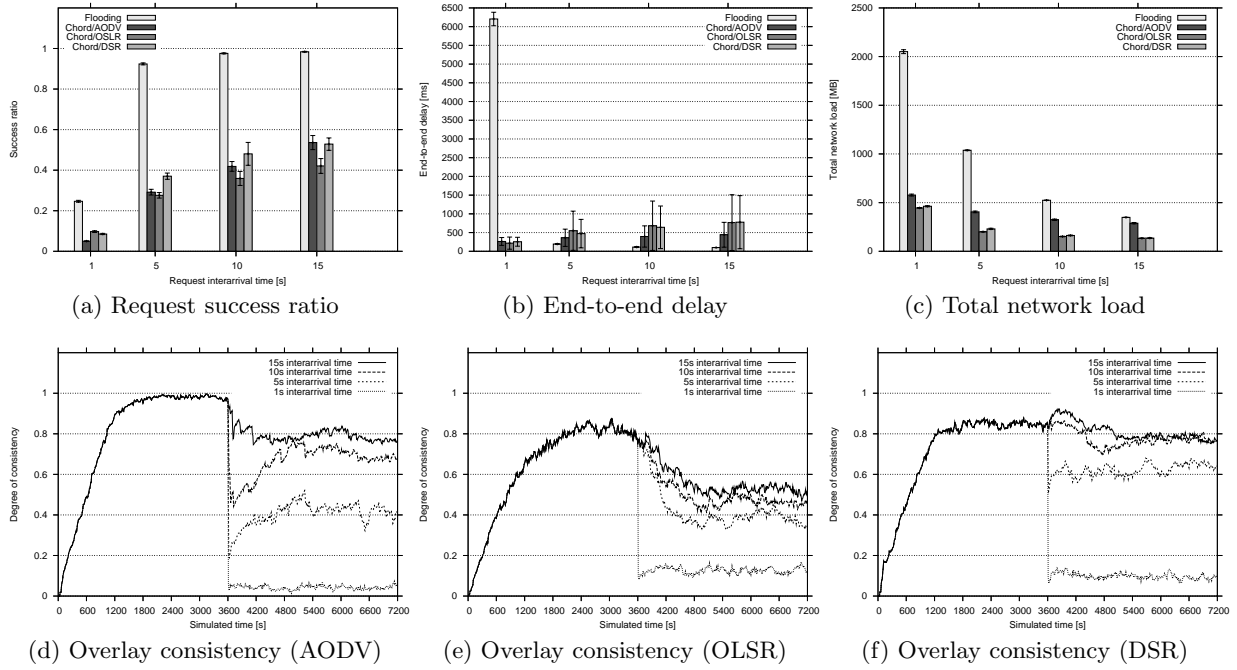


Figure 1: Performance figures for varying offered load

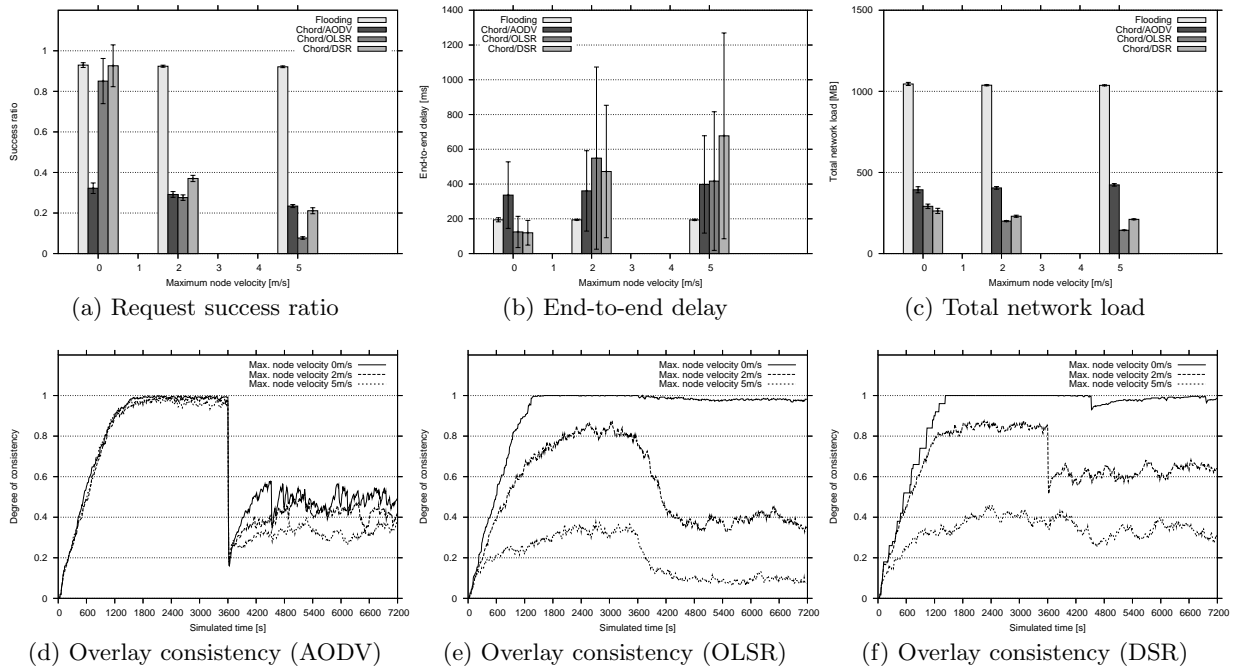


Figure 2: Performance figures for varying node mobility

2(a)). When varying the mobility, Chord exhibits the same inconsistency problems as in the load test (see 2(d)–2(f)), and the success ratio drops to 10–20% for the highest mobility. In static networks, Chord over OLSR and DSR are able to sustain a high degree of consistency leading to approximately 90% successful requests. AODV is unable to handle the relatively high traffic load.

With flooding, the end-to-end delay of a request is approximately 200 ms (Fig. 2(b)), regardless of the amount of mobility. In Chord, the delays and their variability increase with the mobility. The total traffic load of flooding is ca. 1040 MB (Fig. 2(c)) for all examined degrees of node mobility. For Chord, it is one third of that amount.

3.2.4 Varying Network Size

For the last set of experiments, we varied the network size between 10 and 100 nodes in increments of 10. The maximum node speed was 2 m/s, and each node sent requests every 5 s on average. Note that in our model, each node in the network is a traffic source and the offered load therefore also increases with the network size.

With an increasing network size, the success ratio of both flooding and Chord degrades from 100% with 10 nodes to ca. 40% and 10% for 100 nodes, respectively (Fig. 3(a)). Again, Chord suffers from reduced consistency (see Figs. 3(d)–3(f)). With flooding, the end-to-end delay steeply increases from 27 ms to 5 s, but only slowly with Chord, for example from 37 ms to 634 ms in the case of DSR. We observe a similar behavior for the traffic load, which rises from 42 MB to 3.1 GB in the case of flooding (Fig. 3(c)), and from 21 MB to 503 MB in the case of Chord over DSR.

3.3 Discussion

The close connection between the increase in offered load and the decrease in the number of successfully resolved lookups at first is a sign of congestion collapse (cf. Sec. 3.2.2). This conclusion would support the theory of a prohibitive cost for running an overlay protocol over a MANET routing protocol, but is misleading.

Congestion manifests itself in further metrics. In a congestion situation, both the end-to-end delay and the total network load steeply increase. For example, the flooding protocol generates an exponentially increasing overall traffic load with an increasing offered load (Fig. 1(c)). Queue lengths and therefore delays increase as a result of the frequently blocked channel (Fig. 1(b)). At an average request rate of one per node per second, the network shows clear signs of congestion: the success ratio is low, network load is high, and the end-to-end delay increases one order of magnitude. The same signs occur when increasing the network size (Sec. 3.2.4), but not when increasing the degree of node mobility (Sec. 3.2.3), as flooding is a robust protocol that does not react to a dynamic network topology.

Chord does not exhibit signs of congestion. For all tested offered loads, the traffic load experienced by the network is significantly less than with the flooding protocol. The average end-to-end delay of a request does not increase with a higher traffic load (Fig. 1(b)). To further reconstruct the cause of the low request success ratios with Chord, we calculated the total fraction of lookups that are successfully completed, regardless of whether they are consistently or inconsistently resolved. For all investigated traffic loads, degrees of mobility, and network sizes, the requests completed

with a probability of more than 90%.

Thus, we can conclude that in contrast to the prevalent opinion in the literature, the low lookup success ratios observed in Chord are not the result of congestion-related packet loss. They are the result of inconsistencies caused by packet loss due to *mobility or transmission errors*. Chord’s pessimistic failover strategy immediately discards state upon a lookup forwarding timeout. The total number of forwarding failures and consequently the frequency of successor invalidation increases with the number of requests sent. Network consistency drops and subsequent lookups are incorrectly forwarded. The stabilization protocol keeps consistency at a level which is dependent on the rate of its execution and on the successor failure rate.

In the Internet, the failover strategy is used for the early detection of node departures. To mitigate the effects of temporary inconsistency, objects are replicated at multiple successors of the object key. Replication, however, does not solve the problem of network inconsistency, and is expensive in a MANET.

4. CONCLUSION

The layered deployment of structured P2P protocols in a MANET is commonly deemed to be inefficient due to the overhead paid for the separation between the virtual and physical networks. In this ongoing discussion, few quantitative arguments to support this claim have been published.

We simulated Chord in deployments over AODV, DSR, and OLSR. In the majority of all tested scenarios, which varied in network size, node mobility, and offered application load, we found that Chord’s ability to consistently resolve lookups was significantly impaired. Notwithstanding, the observed performance issues unexpectedly were *not* a result of congestion. Upon packet loss due to mobility or erroneous transmission, the protocol’s failover strategy prematurely assumes that the packet’s destination departed from the network. It disposes of the corresponding state information and thereby lowers network consistency. Lookups are inconsistently resolved, resulting in incorrect application behavior.

5. REFERENCES

- [1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GloMoSim: A Scalable Network Simulation Environment. Technical Report 990027, UCLA Computer Science Department, 1999.
- [2] L. Barbosa e Oliveira, I. G. Siqueira, and A. A. F. Loureiro. Evaluation of Ad-Hoc Routing Protocols under a Peer-to-Peer Application. In *Proc. IEEE WCNC 2003*, pages 1143–1148, New Orleans, LA, USA, Mar. 2003.
- [3] E. Borgia, M. Conti, F. Delmastro, and E. Gregori. Experimental Comparison of Routing and Middleware Solutions for Mobile Ad Hoc Networks: Legacy vs Cross-Layer Approach. In *Proc. ACM SIGCOMM ’05 Workshops*, pages 82–87, Philadelphia, PA, USA, Aug. 2005.
- [4] J. Broch, D. B. Johnson, and D. A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, Oct. 1999. Internet draft version 3.
- [5] M. Caesar, M. Castro, E. B. Nightingale, G. O’Shea, and A. Rowstron. Virtual Ring Routing: Network

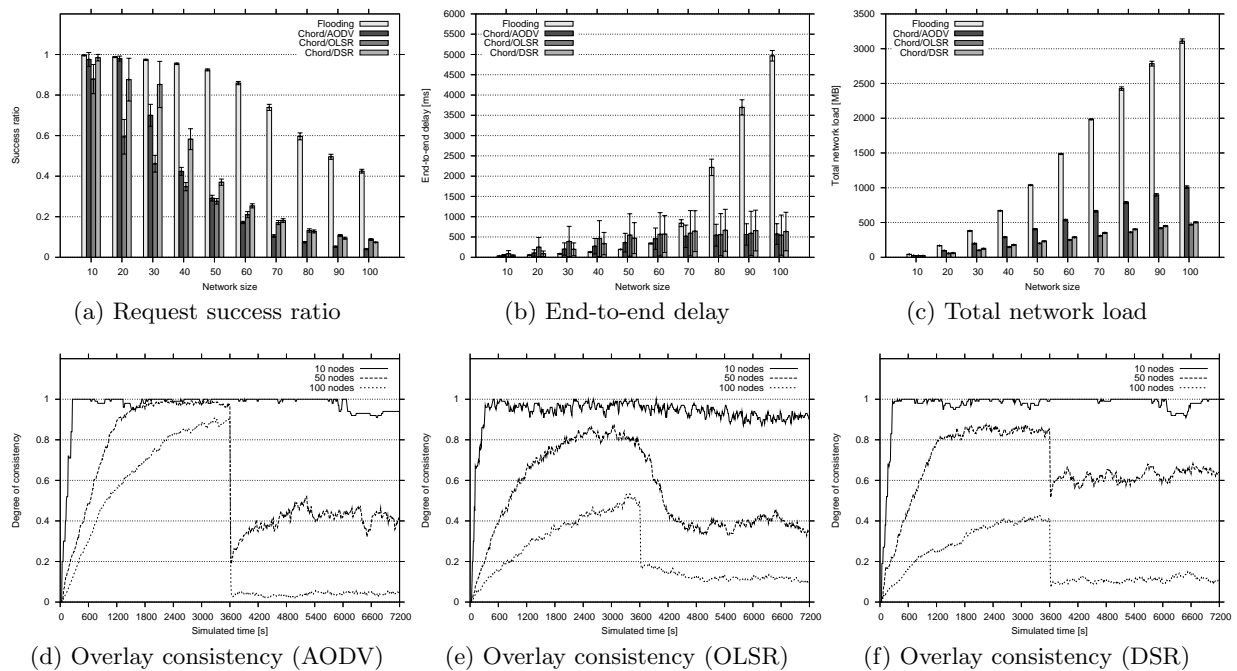


Figure 3: Performance figures for varying network size

- Routing Inspired by DHTs. In *Proc. ACM SIGCOMM '06*, Pisa, Italy, Sept. 2006. To appear.
- [6] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol, July 2003. Internet draft version 11.
- [7] C. Cramer and T. Fuhrmann. ISPRP: A Message-Efficient Protocol for Initializing Structured P2P Networks. In *Proc. 24th IPCCC*, pages 365–370, Phoenix, AZ, USA, Apr. 2005.
- [8] C. Cramer and T. Fuhrmann. Bootstrapping Chord in Ad Hoc Networks: Not Going Anywhere for a While. In *Proc. IEEE MP2P '06*, Pisa, Italy, Mar. 2006.
- [9] J. Eberspächer, R. Schollmeier, S. Zöls, and G. Kunzmann. Structured P2P Networks in Mobile and Fixed Environments. In *Proc. HET-NETs '04*, Ilkley, West Yorkshire, UK, July 2004.
- [10] T. Fuhrmann, P. Di, K. Kutzner, and C. Cramer. Pushing Chord into the Underlay: Scalable Routing for Hybrid MANETs. Technical Report 2006-12, Universität Karlsruhe (TH), Germany, Fakultät für Informatik, 2006.
- [11] I. Gruber, R. Schollmeier, and W. Kellerer. Performance Evaluation of the Mobile Peer-to-Peer Service. In *Proc. IEEE CCGrid 2004*, pages 363–371, Chicago, IL, USA, Apr. 2004.
- [12] Y. C. Hu, S. M. Das, and H. Pucha. Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks. In *Proc. HotOS IX Workshop*, pages 37–42, Lihue, HI, USA, 2003.
- [13] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proc. IEEE INMIC*, pages 62–68, Lahore, Pakistan, Dec. 2001.
- [14] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, 353:153–181, Feb. 1996.
- [15] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, Feb. 2003. Internet draft version 13.
- [16] C. E. Perkins and E. M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proc. 2nd IEEE WMCSA*, pages 90–100, New Orleans, LA, USA, Feb. 1999.
- [17] H. Pucha, S. M. Das, and Y. C. Hu. Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks. In *Proc. 6th IEEE WMCSA*, English Lake District, UK, Dec. 2004.
- [18] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Middleware 2001*, Heidelberg, Germany, Nov. 2001.
- [19] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, Feb. 2003.
- [20] L. Yan, K. Sere, and X. Zhou. Towards an Integrated Architecture for Peer-to-Peer and Ad Hoc Overlay Network Applications. In *Proc. IEEE FTDCS '04*, pages 312–318, Suzhou, China, May 2004.
- [21] T. Zahn and J. Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. In *Proc. 5th ASWN 2005*, Paris, France, June 2005.